Amendments to the Claims:

A listing of the entire set of pending claims (including amendments to the claims, if any) is submitted herewith per 37 CFR 1.121. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Previously presented) A method for training a self ordering map for use in a computing system, comprising:

initializing a set of weights of the self ordering map; and

iteratively training the weights over many training epochs;

wherein

for at least a number of the training epochs, iteratively training the weights includes

updating the weights based on a learning rate that is generated according to a function that changes in a fashion that is other than monotonically decreasing with the training epochs.

2. (Previously presented) A method as in claim 1, wherein

the function includes a random or pseudorandom function.

3. (Previously presented) A method as in claim 2 wherein

the random or pseudorandom function has a range that decreases with the training epochs.

4. (Previously presented) A method as in claim 2 wherein

the random or pseudorandom function is configured such that the learning rate tends to decrease with the training epochs.

5. (Previously presented) A method as in claim 1 wherein

the function has a range that decreases with the training epochs.

6. (Previously presented) A method as in claim 5 wherein

the function is configured such that the learning rate tends to decrease with the training epochs.

7. (Previously presented) A method as in claim 1 wherein

the function is configured such that the learning rate tends to decrease with the training epochs.

8. (Previously presented) A method of training a self ordering feature map for use in a computing system, comprising:

choosing a random value for initial weight vectors;

drawing a sample from a set of training sample vectors and applying it to input nodes of the self ordering feature map;

identifying a winning competition node of the self ordering feature map according to a least distance criterion;

adjusting a synaptic weight of at least the winning node, using a learning rate to update the synaptic weight that is based on a function other than one that is monotonic with subsequent training epochs;

iteratively repeating the drawing, identifying, and adjusting to form each subsequent training epoch.

9. (Previously presented) A method as in claim 8, wherein

the function corresponds to a random or pseudorandom function.

10. (Previously presented) A method as in claim 9 wherein

the function has a range that decreases with subsequent training epochs.

11. (Previously presented) A method as in claim 9 wherein

the function is configured such that the learning rate tends to decrease with subsequent training epochs.

12. (Previously presented) A method as in claim 8 wherein

the function has a range that decreases with subsequent training epochs.

13. (Previously presented) A method as in claim 12 wherein

the function is configured such that the learning rate tends to decrease with subsequent training epochs.

14. (Previously presented) A method as in claim 8 wherein

the function is configured such that the learning rate tends to decrease with subsequent training epochs.